بسم الله الرحمن الرحيم

قال تعالى " و قل ربي زدني علما "

الخوارزميات الجزء الأول

نبدأ بتعريف الخوارزمية بشكل عام:

الخوارزمية (algorithm) هي مجموعة من الخطوات أو العمليات المنطقية التي تؤدي لحل مسألة ما.

المثال الأول: برنامج نظام الانتخاب

لنفرض وجود عدد n من المرشحين نريد أن نحسب عدد أصوات الناخبين لكل منهم و طباعة عدد الأصوات لكل مرشح

الحل:

```
Const n=20;
Index:1..20;
Type mat=arrau[index] of integer;
Procedure voter(var a:mat);
Var i:integer;
Begin
For i:=1 to n do
A[i]:=0;
Readln(i);
While(i<>0)do
Begin
If (1<=i) and (i<=n)then
A[i]:=a[i]+1;
Readln(i);
End;
For i:=1 to n do
Writeln('mat ',I,'=',a[i]);
End;
```

المثال الثاني: باستخدام بنية المصفوفات إجرائية إظهار عدد صحيح بشكل معكوس

N=1234 ⟸ N=4321

```
Type mat=array[1..10] of integer;
Procedure reverse (a:mat; n:integer);
Var I,j:integer;
Begin
I:=0;
While(N<>0)do
Begin
I:=i+1;
A[i]:=n mod 10;
N:=n div 10;
End;
For j:=1 to i do
Write(a[j]);
End;
Begin
reverse(a,N);
end.
```


اللهم صلي و سلم و بارك على سيدنا محمد و على آله و صحبه أجمعين

المثال الثالث: إجرائية تكرارية تحول من عدد عشري إلى عدد ثنائي

```
Procedure bin2dec(x:integer);
Var I,k:integer;
Begin
I:=1;
While(x<>0)do
Begin
A[i]:=x mod 2;
X:=x div 2;
i:=i+1;
end;
for k:=i-1 downto 1 do
write(a[k]);
end;
```

المثال الرابع: إجراء تكراري يحول من ثنائي إلى عشري

2 1 0

$101=1*2^0+0*2^1+1*2^2=1+0+4=(5)_{10}$

```
Function dec2bin(x:integer):integer;
Var res, i:integer;
Begin
Res:=0; i:=0;
While(x<>0)do
Begin
Res:=res+ x mod 10* power(i);
X:=x div 10; i:=i+1;
End;
Dec2bin:=res;
End;
```

المثال الخامس: المصفوفات

مصفوفة مربعة n=4

*			
*	*		
*	*	*	

الخوارزمية تقوم بحساب مجموع عناصر ما تحت القطر الرئيسي (المثلث السفلي)للمصفوفة المربعة

1

```
Mat[100][100]:integer;
n,i,j,x1:integer;
read(n);
x1 ←0
i ← 1
while(i≤n)
j ← 1
while(j<1)
x1 ← x1+mat[i][j];
j:=j+1;
i ← i+1;
write(x1);
```

$\sum_{i=1}^{n}\sum_{j=1}^{i-1}1 = \sum_{i=1}^{n} i - 1 = \sum_{i=1}^{n} i - \sum_{i=1}^{n} 1 = \frac{n(n+1)}{2} - n = \frac{n^2-n}{2}$=o($n^2$)

إذاً عدد عمليات الجمع هي $\frac{n^2-n}{2}$ و درجة العقيد هي من مرتبة n^2.

المثال السادس: المصفوفات

مصفوفة مربعة n=4

*	*	*	
*	*	*	
*	*	*	
*	*	*	

2

```
Mat[100][100]:integer;
n,i,j,x2:integer;
readn(n);
x2 <— 0
I <— 1
While(i≤n)
j <— 1
while(j≤3)
x2 <— x2+mat[i][j];
j <— j+1;
I <— i+1;
Write(x2);
```

الخوارزمية تقوم بحساب مجموع عناصر الأعمدة الثلاثة الأولى من المصفوفة

$$\sum_{i=1}^{n}\sum_{j=1}^{3}1=\sum_{i=1}^{n}3=3n=o\left(n^2\right)$$

إذاً عدد عمليات الجمع هي 3n ودرجة التعقيد من مرتبة n^2.

المثال السابع: المصفوفات

مصفوفة مربعة n=4

			*
		*	
	*		
*			

3

```
Mat[100][100]:integer;
n,i,j,x3:integer;
read(n);
x3 <— 0
j <— n
while(j≥1)
x3 <— x3+mat[n-j+1][j];
j <— j-1;
write(x3);
```

الخوارزمية تقوم بحساب مجموع عناصر القطر الثانوي للمصفوفة

$$\sum_{j=1}^{n}1=n=o\left(n\right)$$

إذاً عدد عمليات الجمع هي n ودرجة التعقيد من مرتبة n.

المثال الثامن:

مصفوفة مربعة n=4

	*	*	*
		*	*
			*

4

```
Mat[100][100]
n,i,j,x4:integer;
read(n);
x4 <— 0
i <— 1
While(i≤n)
j <— n
while(j>i)do
x4 <— x4+mat[i][j];
j <— j-1;
I <— i+1;
Write(x4);
```

الخوارزمية تقوم بحساب مجموع عناصر ما فوق القطر الرئيسي(المثلث العلوي للمصفوفة المربعة)

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(1)=\sum_{i=1}^{n}n-i=\sum_{i=1}^{n}n+\sum_{i=1}^{n}i=n^2-\left[\frac{n(n+1)}{2}\right]=\frac{n2-n}{2}=o(n^2)$$

ملاحظة: $\sum_{j=1}^{n}1=\sum_{j=1}^{i}1+\sum_{j=i+1}^{n}1$

إذاً عدد عمليات الجمع هو $\frac{n^2-n}{2}$ ودرجة التعقيد من مرتبة n^2.

المثال التاسع: حساب القاسم المشترك الأكبر لعددين صحيحين موجبين:

```
Unsigned  int gcd(Unsigned  int m, Unsigned  int n){
Unsigned  int rem;
While(n>0){
rem=m%n;
m=n;
n=rem;
}
return (m);
}
```

المثال العاشر:حساب القاسم المشترك الأكبر لعددين صحيحين موجبين بطريقة **إقليدس**:

```
Function gcd (m,n:integer):integer;
Begin
If(m=n)then
gcd:=m
else
begin
if(m>n)then
m:=m-n
else
n:=n-m;
gcd:=(m,n);
end;
end;
```

المثال الحادي عشر:مصفوفات

لتكن لدينا مصفوفة M[1..n][1..n] من الأعداد الطبيعية تحتوي على n^2 عنصر بلا أي ترتيب تريد تحويل هذه المصفوفة إلى مصفوفة مفروزة وفق الفرز التالي:

- يجب أن يكون العناصر مرتبة في كل سطر:

M[x][y]< M[x][y+1] all x and y

- يجب أن تكون العناصر مرتبة في كل عمود:

M[x][y]< M[x+1][y] all x and y

اكتب إجرائية فرز المصفوفة وفق الترتيب السابق وما هو تعقيد الإجرائية.

الحل:

بنية المعطيات هي كالتالي:

```
Type matrix=array[1..n,1..n] of real;
```

إجرائية فرز المصفوفة في كل سطر:

```
Procedure sortline(var m:matrix; n:integer);
Var k,i,j:integer;
Begin
For k:=1 to n do
For i:=1 to n do
For j:=1 to n-1 do
If(m[i,j]>m[I,j+1])then
Begin
Temporary := m[i,j];
m[i,j]:=m[[i,j+1];
m[i,j+1]:=temporary;
end;
```

```
end;
```

$$\sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n-1}1=\sum_{k=1}^{n}\sum_{i=1}^{n}n-1=\sum_{k=1}^{n}n*(n-1)=n^3-n^2=O(n^3)$$

إجرائية فرز المصفوفة في كل عمود :

```
Procedure sorthorizontal(var m:matrix; n:integer);
Begin
For k:=1 to n do
For i:=1 to n-1 do
For j:=1 to n do
If(m[i,j]>m[i+1,j])then
Begin
Temporary := m[i,j];
m[i,j]:=m[[i+1,j];
m[i+1,j]:=temporary;
end;
end;
```

$$\sum_{k=1}^{n}\sum_{i=1}^{n-1}\sum_{j=1}^{n}1=\sum_{k=1}^{n}\sum_{i=1}^{n-1}n=\sum_{k=1}^{n}n*(n-1)=n^3-n^2=O(n^3)$$

المثال الثاني عشر: تعقيد الخوارزميات

ليكن لدينا التابع : والمطلوب ايجاد تعقيده بدلالة n؟

```
p = 1;
for(int i = 1; i <n; i++)
{
  p = p * n;
}
```

- نلاحظ بأنه لدينا حلقة واحدة تمتد من 1 إلى n
- عند كل دخول بهذه الحلقة نقوم بتنفيذ عملية ضرب واحدة
- فيكون التعقيد على الشكل:

$$\sum_{i=1}^{n}1=O(n)$$

المثال الثالث عشر: تعقيد الخوارزميات

ليكن لدينا التابع :

```
int res = 1;
factor = x;
while (n>0)
{
   if((n mod 2) == 1)
     res = res * factor;
   factor = factor * factor;
   n = n div 2;
}
```

المطلوب: إيجاد تعقيد التابع بدلالة n:

- تبدأ الحلقة بالقيمة n وعند كل مرور نقسم على 2

- عند المرور الأول يكون حجم المسألة : $\frac{n}{1}$
- عند المرور الثاني $\frac{n}{2}$
- عند المرور الثالث $\frac{n}{4}$
- عند المرور رقم i تكون يكون بعد المسألة من مرتبة $\frac{n}{2^{i-1}}$
- على فرض أن j هو المرور الأخير ضمن الحلقة السابقة :
- تكون قيمة n مساوية للصفر، عندها ولكي نتمكن من حساب قيمة j يتوجب علينا ايجاد حل للمعادلة لوغارتمية التالية:

$$\frac{n}{2^{j-1}}=1 \Rightarrow n=2^{j-1} \Rightarrow$$ نأخذ لوغارتم الطرفين

$$\log_2(n)=\log_2(2^{j-1}) \Rightarrow \log_2(n)=(j-1)*\log_2(2) \Rightarrow \log_2(n)=(j-1)$$

- تكون قيمة j كالتالي:

$$j=\log_2(n)+1$$

- يوجد عمليتي ضرب ضمن الحلقة و عملية قسمة ،عمليات الضرب و القسمة التي يتم تنفيذها ضمن كل دخول بالحلقة هو 3 عمليات
- بالتالي يكون تعقيد الخوارزمية كالتالي: $Cost=3*(\log_2(n)+1)=3\log_2(n)+3$

$$\Rightarrow Cost=O(\log_2(n))$$

المثال الرابع العشر: تعقيد الخوارزميات

ليكن لدينا التابع التالي و المطلوب: إيجاد تعقيد التابع بدلالة n:

```
for i = 1 to m
  for j = 1 to q
    for k = 1 to n
      x = x * 2;
```

- كلفة الحلقة الداخلية : $\sum_{k=1}^{n} 1$
- كلفة الحلقة الوسطى: $\sum_{j=1}^{q}\sum_{k=1}^{n} 1$
- كلفة الحلقة الخارجية: $\sum_{k=1}^{m}\sum_{j=1}^{q}\sum_{i=1}^{n} 1$

$$Cost=\sum_{k=1}^{m}\sum_{j=1}^{q}\sum_{i=1}^{n} 1 \Rightarrow Cost=\sum_{k=1}^{m}\sum_{j=1}^{q} n \Rightarrow Cost=\sum_{k=1}^{m} q*n$$

$$\Rightarrow Cost=m*q*n$$

- و على فرض بأن m=q=n يكون التعقيد أو عدد العمليات الناجم عن تنفيذ هذه الخوارزمية هو n*n*n

$$\Rightarrow Cost=O(n^3)$$

إذاً تعقيد الخوارزمية هو من مرتبة n^3.

المثال الخامس عشر: تعقيد الخوارزميات

ليكن لدينا الخوارزمية التالية:و المطلوب إيجاد تعقيد التابع بدلالة n:

```
for i = 1 to n-1
  for j = i+1 to n
    for k = 1 to j
    {
      m = m + 1;
                p = p* 2;
                l = l +3;
```

}

سوف نركز فقط على عمليات الضرب وذلك لأن كلفتها مرتفعة مقارنة بعملية الجمع

كلفة الحلقة الداخلية: $\sum_{k=1}^{j} 1$

كلفة الحلقة الوسطة: $\sum_{j=i+1}^{n}\sum_{k=1}^{j} 1$

كلفة الحلقة الخارجية : $\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{j} 1$

$$\sum_{1}^{n} = \sum_{1}^{i} + \sum_{i+1}^{n} \Rightarrow \sum_{i+1}^{n} = \sum_{1}^{n} - \sum_{1}^{i}$$

$$Cost = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{j} 1 = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} j$$

$$Cost = \sum_{i=1}^{n-1}\left(\sum_{1}^{n} j - \sum_{1}^{i} j\right) \Longrightarrow Cost = \sum_{i=1}^{n-1}\left(\frac{n(n+1)}{2} - \frac{i(i+1)}{2}\right) \Rightarrow$$

$$Cost = \frac{1}{2}\sum_{i=1}^{n-1} n^2 + \frac{1}{2}\sum_{i=1}^{n-1} n - \frac{1}{2}\sum_{i=1}^{n-1} i^2 - \frac{1}{2}\sum_{i=1}^{n-1} i \Rightarrow Cost \approx O(n^3)$$

إذا درجة التعقيد من مرتبة n^3.

المثال السادس عشر: تعقيد الخوارزميات

ليكن لدينا الخوارزمية التالية:والمطلوب إيجاد تعقيد التابع بدلالة n:

```
for i=1 to n
 if( odd(i) == true)
 begin
   for j=1 to n
     x = x +1;
   for j=1 to i
     y = y*2;
 end;
```

نلاحظ بأنه يوجد حلقتين داخل الشرط ، تقوم الحلقة الأولى بتنفيذ عمليات جمع بينما تقوم الحلقة الثانية بتنفيذ عمليات ضرب لذا سوف نقوم بحساب تعقيد عمليات الجمع على حدة ، ثم حساب تعقيد عمليات الضرب ويكون تعقيد الخوارزمية هو مجموع كل من التعقيدين.

عمليات الجمع:

عند كل دخول في حلقة الجمع نقوم بإجراء n عملية جمع.

يكون تعقيد عمليات الجمع على الشكل التالي:

$$= \sum_{i=1}^{n} \Big\|^{odd(i)} \sum_{j=1}^{n} 1 = \sum_{i=1}^{n} \Big\|^{odd(i)} n$$

بما أن عدد الأعداد الفردية من 1 إلى n هو $\frac{n+1}{2}$.

$$= \sum_{i=1}^{n} \Big\|^{odd(i)} n = n * \frac{n+1}{2} \Longleftarrow$$

ويكون التعقيد الناتج على الشكل التالي: $O(n^2)$.

عمليات الضرب:

$$\sum_{i=1}^{n} \Big\|^{odd(i)} \sum_{j=1}^{i} 1$$

$$=\sum_{i=1}^{n} \|^{odd(i)} i$$

=(1 + 3 + 5 + 7 + 9 + ... + 2m+1) : 2m+1 <n

أي المجموع المطلوب هو مجموع الأعداد الفردية ويكافئ مجموع جميع الأعداد – مجموع الأعداد الزوجية

$$\sum_{i=1}^{n} \|^{odd(i)} \sum_{j=1}^{i} 1$$

$$=\sum_{i=1}^{n} \|^{odd(i)} i$$

$$=\frac{n(n+1)}{2}-\frac{n}{2}(\frac{n}{2}+1)$$

$$\approx O(n^2)$$

المثال السابع عشر: تعقيد الخوارزمية

ليكن لدينا الخوارزمية التالية:المطلوب احسب تعقيد هذا التابع في أسوأ الأحوال بدلالة n:

```
Prod:=1;
Nfor := sqr(n) * sqr(n);
for k:=1 to Nfour do
  if k mod sqr(n) = 0 then
    for j:=1 to k do
      if j mod sqrt(n) = 0 then
        for m:=1 to j do
          prod:= prod *4;
```

- يتم تنفيذ الحلقة الداخلية لوحدها j مرة وفي كل مرة تقوم بإجراء عملية ضرب واحدة .
- تكون كلفة الحلقة الداخلية : $Cost(Loop_3)=\sum_{m=1}^{j} 1$
- لا يتم تنفيذ محتوى الحلقة الوسطى إلا بشرط على العداد j وهو أن يقبل القسمة على $\sqrt{n}$
- بذلك تكون كلفة الحلقة الوسطى بدلالة حدها الأعلى k هي $Cost_{loop2}=\sum_{j=1}^{k} \|_{cond}\ (j)$

$$Cost_{loop2}=\sum_{j=1}^{k} \|_{cond}\ (j) \Rightarrow Cost_{loop2}=\sum_{j`=1}^{\frac{k}{\sqrt{n}}} \sqrt{n}.j` = \sqrt{n}.\sum_{j`=1}^{\frac{k}{\sqrt{n}}} j`$$

$$\Rightarrow Cost_{loop2}=O(\frac{k^2}{\sqrt{n}})$$

- أما بالنسبة للحلقة الأولى فلا يتم تنفيذ محتواها إلا بشرط على العداد k وهو أن يقبل القسمة على العدد n^2.
- عند تحقق الشرط يتم تنفيذ عدد من العمليات من مرتبة $\frac{k^2}{\sqrt{n}}$ وبذلك تكون كلفتها بدلالة حدها الأعلى n^4 على الشكل التالي:

$$Cost_{loop1}=\sum_{k`=1}^{\frac{n^4}{n^2}} \frac{(n^2.k`)^2}{\sqrt{n}} \Leftarrow Cost_{loop1}=\sum_{k=1}^{n^4} \|_{n^2|k}\ (\frac{k^2}{\sqrt{n}})$$

- يتم تنفيذ الحلقة الداخلية لوحدها j مرة وفي كل مرة تقوم بإجراء عملية ضرب واحدة .
- تكون كلفة الحلقة الداخلية: $Cost(Loop_3)=\sum_{m=1}^{j} 1$

- بالتالي يكون تعقيد الخوارزمية على الشكل التالي: $Cost_{loop1} = \frac{n^4}{\sqrt{n}}.\sum_{k`=1}^{n^2} k`^2$

$$\Longrightarrow \quad Cost_{loop1} = \frac{n^4}{\sqrt{n}}.O(n^2)^3 = n^{3.5}O(n^6) = O(n^{9.5})$$

المثال الثامن عشر: تعقيد الخوارزميات

ليكن لدينا التابع التالي:

```
Function strange(x,n:integer):integer;
Var i,j,px,sum:integer;
Begin
i:=2; sum:=x;
while(i<=n)do
begin
j:=2; px:=x;
while(j<=n)do
begin
px:=px*x; j:=j+1;
end;
sum:=sum+px; i:=i+1;
end;
strange:=sum;
end;
```

1 - ما الذي يحسبه التابع
2 - ما هي درجة تعقيده
3 - أعد كتابة هذا التابع بحيث يحسب القيمة نفسها و لكن بعدد عمليات اقل
4 - ما هي درجة تعقيد التابع الجديد

الحل:

التابع يقوم بحساب:

$$\sum_{i=1}^{n} x^i = x^1 + x^2 + \dots\dots\dots + x^n$$

التابع يقوم بحساب مجموع رفع لقوة i من 1 إلى n بالنسبة للعدد x

درجة تعقيد التابع:

$$\sum_{i=2}^{n}\sum_{j=2}^{i} 1 = \sum_{i=2}^{n} i - 1 = \frac{n^2 - n}{2} \simeq O(n^2)$$

كتابة التابع السابق بعدد عمليات أقل:

```
Function strange (x,n:integer):integer;
Var
Begin
 px:=1; sum:=0;
for i:=1 to n do
begin
px:=px*x; sum:=sum+px;
end;
strange:=sum;
```

```
end;
```

درجة تعقيد التابع الجديد:

$$\sum_{i=1}^{n} 1 = n \simeq O(n)$$

(و ما توفيقي إلا بالله)